

COMPARATIVOS DE AMBIENTES DE PROGRAMAÇÃO GRÁFICOS PARA O KIT DE ROBÓTICA LEGO MINDSTORMS NXT

Brenda Kieling Balbinotti¹, Nicole Dalmolin Kochan². Marcelo Massocco Cendron³

Instituto Federal Catarinense de Ciência, Computação e Tecnologia – *Campus Videira* (IFC)

Videira – SC – Brasil

brenda__kb¹, nicole_kochan²@hotmail.com, marcelo.cendron@ifc-videira.edu.br³

RESUMO

Esse artigo propõe a comparação de três ambientes de programação diferentes aplicados para robótica. De forma a ajudar o leitor em qual ambiente irá utilizar, esse artigo visa apresentar algumas características desses ambientes e alguns pequenos exemplos de programação utilizando seus blocos de comandos gráficos disponíveis.

Palavras-chave: Robótica, Ambiente, Programação,

1 INTRODUÇÃO

A programação de robôs incluiu projetar diferentes formas de controlar seus movimentos de acordo com suas características construtivas. Nesse sentido, o equipamento físico, assim com a forma de programação na área de robótica tem demonstrado uma crescente evolução com fins didáticos, percebe-se avanços na robótica com frentes em simulação, no desenvolvimento de kits de robótica de fácil montagem e em linguagens de programação extremamente amigáveis para quem não tem conhecimento aprofundado de programação. Como exemplo o kit LEGO MINDSTORM NXT, lançado em 2006. Sua utilização tem sido bem sucedida devido a sua fácil manipulação e pela sua versatilidade quando se refere às formas de programação disponibilizadas no kit. O ambiente de programação do LEGO MINDSTORMS NXT é o NXT 2.0 Programming, que utiliza blocos de comandos bem estruturados e de fácil compreensão, principalmente para

usuários que estão iniciando sua interação com o kit ou programações em outros formatos. [1]

Porém existem outros ambientes de programação e que propiciam uma maior diversidade de funções que podem ser exploradas do kit. Pode-se destacar a utilização do software pago MATLAB, um ambiente de programação matemática que possui uma vasta disponibilidade de recursos que podem ser utilizadas em outras áreas, e a VPL (Visual Programming Language), programação gráfica baseado em fluxo de dados que contém ainda a VSE, plataforma do simulador, que juntas constituem as plataformas da Microsoft Robotics Developer Studio (MSRS).[1]

O objetivo desse artigo é analisar e comparar as diferenças e semelhanças existentes entre três linguagens citadas acima, com o intuito de ter resultados que possam facilitar a compreensão.

2 PROCEDIMENTOS METODOLÓGICOS

De forma a permitir uma análise comparativa sobre as linguagens de programação gráficas aplicadas para o LEGO MINDSTORMS, procuramos detalhar inicialmente o kit didático de robótica.

2.1 O kit LEGO MINDSTORMS NXT versão Educacional.

O kit é composto de um bloco com entradas para os sensores e saídas para os motores, um sensor ultrassônico, um sonoro, um de luz e dois de toque. Estão inclusas também várias peças características dos brinquedos Lego, que possibilitam a montagem de diversos modelos de robôs e máquinas.

A conexão do NXT com o computador pode ser feita via USB ou Bluetooth, este último, apesar da comunicação se tornar mais lenta, ele proporciona uma liberdade no deslocamento dos robôs. [3]

A figura 1 apresenta as partes que compõem o Kit LEGO MINDSTORMS NXT.

Figura 1 - Kit LEGO MINDSTORMS NXT.



Fonte: [2]

Existem dois softwares básicos que acompanham o kit, o NXT Programming e o Data Logging

2.1.1 NXT 2.0 Programming

Apresenta um ambiente de programação de alto nível, onde eventos que irão acontecer ao longo da execução do programa, são modelados por blocos de função. Na plataforma do programa são encontrados blocos correspondentes a ações que serão executadas pelos motores do lego que estarão ligados ao brick. Esses sensores e motores podem receber valores medidos e estes podem ser utilizados na lógica do programa. Além dos blocos que executam um evento, podem ser encontrados outros que realizam lógicas.[3]

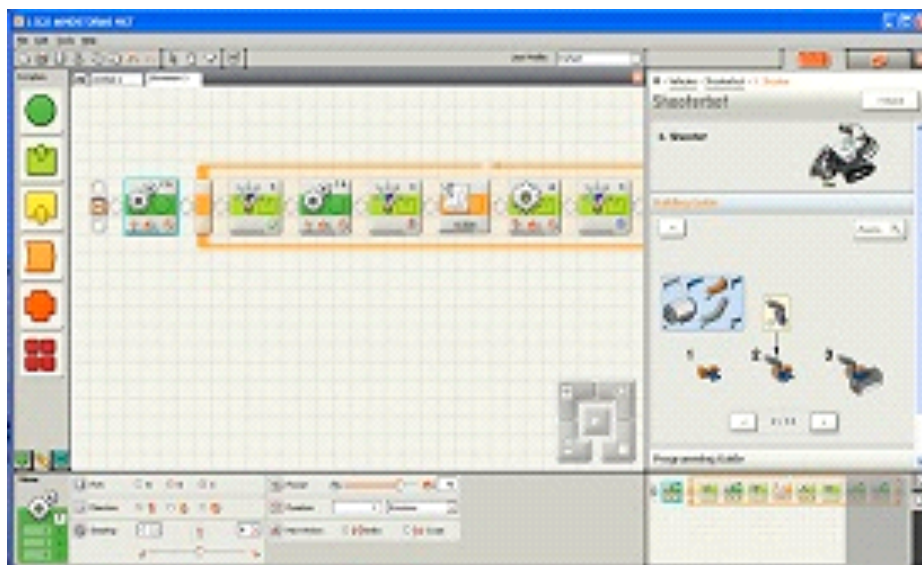
NXT 2.0 Data Logging

Pode ser utilizado para coletar dados. O programa cria um bloco que simula um experimento, onde são definidas as variáveis a serem medidas, o tempo e outros parâmetros. A partir deste programa pode-se verificar desde a variação angular do eixo de um motor até a variação da cor de certa superfície.

Os sensores ultrassônicos podem detectar objetos a até 255 cm de distância. Entretanto, quando o objeto está muito próximo ou em casos de quinas o som pode ser desviado, resultando em erros de medição significativos.

Os motores permitem que se controle o quanto devem girar e a que velocidade podendo variar de 0 a 100, e ainda, possuem sensores de rotação para determinação da distância percorrida dada em graus. [3]

Figura 2 - Diagrama da linguagem NXT.



Fonte: [4]

2.2 MATLAB

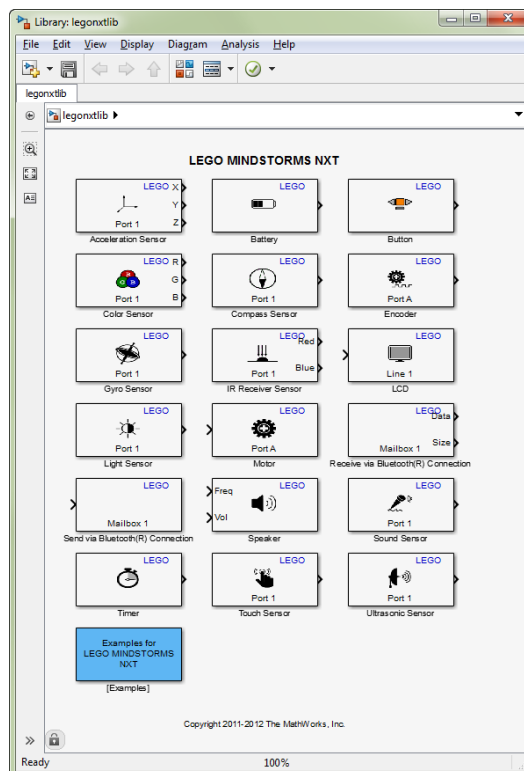
Linguagem de alto nível e um ambiente interativo para computação numérica, visualização e programação. Integrando análise numérica, cálculo com matrizes, processamento de sinais e construção de gráfico em ambiente fácil de usar. A linguagem permite ainda que o programador analise dados, desenvolva algoritmos, crie modelos, crie uma variedade de aplicações incluindo processamento de sinais e comunicação, processamento de imagem e vídeo, sistemas de controle, teste e medição, finanças computacional e biologia computacional. Permitindo explorar abordagens e chegar a uma solução mais rápida do que com planilhas ou linguagens de programação tradicionais C / C++ ou Java. [5]

Dentro das bibliotecas do MATLAB pode-se destacar a Erobot NXT (Embedded Coder Robot NXT), uma biblioteca voltada ao ambiente do Simulink, uma ferramenta para modelagem, simulação e análise de sistemas. Sua interface primária é uma ferramenta de construção de diagramas gráficos por blocos e bibliotecas customizáveis. Este software permite serem realizadas aplicações em tempo real, oferece alta integração com o resto do ambiente.

Amplamente usado em teoria de controle e processamento digital de sinais para projeto e simulação. Porém esta biblioteca possui uma forma de instalação não muito comum e dependente de outros softwares para seu funcionamento, o que pode ser um problema. Permite também a percepção de modelo e sistemas de mobilidade, contendo muitas funções de manipulação do robô, gerar o código C a partir do seu modelo de simulação. A simulação também pode ser usada para conceber e validar seus algoritmos de controle. [5]

Outra possibilidade é o comando do robô utilizando o Bluetooth. Isso proporciona aplicações como controle remoto dos robôs, aquisição de dados, uma vez que os dados dos sensores também podem ser enviados para um PC. [6]

Figura 4 - Gráfico de representação do MATLAB.



Fonte: [7]

2.3 MICROSOFT ROBOTICS DEVELOPER STUDIO (MSRS)

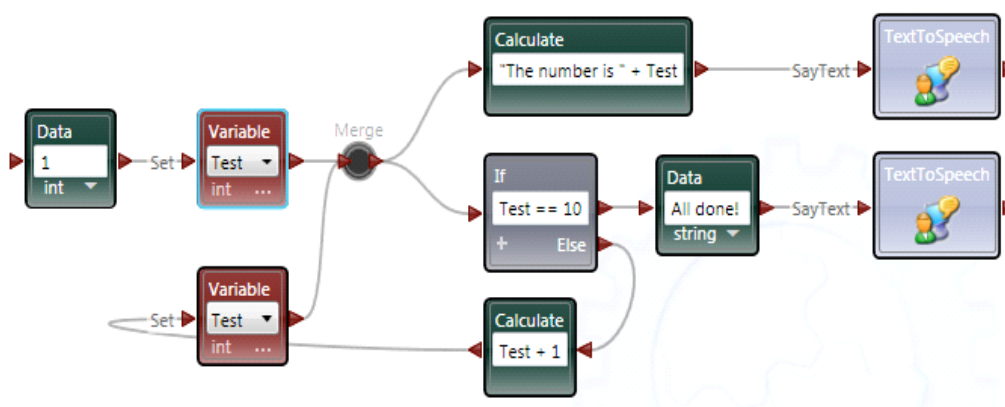
Um ambiente onde é possível desenvolver programas para as aplicações robóticas em diferentes plataformas de hardware. [1] Uma das plataformas Visual Programming Language (VPL), é um ambiente de programação gráfica baseado em fluxo de dados que pode ser utilizado para criar serviços. Os diagramas de fluxos de dados ou programação são baseados em "blocos" ou outros objetos da tela que são tratados como entidades, ligadas por setas, linhas ou arcos que representam as relações. Um fluxo de dados VPL consiste de uma sequência de atividades ligadas representados como blocos com entradas e saídas que podem ser ligadas a outros blocos de atividades. [8]

Outra que pode ser citada é a Visual Simulation Environment (VSE). Utilizada em uma variedade de cenários avançados, visualização e dimensionamento.

O simulador pode ser utilizado para testar algoritmos antes de executá-los no hardware do robô real, evitando que erros de programação que prejudiquem o robô. Além disso, existem casos em que o robô ainda não existe e o simulador funciona como um local de testes para o modelo a ser utilizado. [8]

Existem vários ambientes de simulação no RDS, ambiente de simulação. Esses ambientes foram desenvolvidos por SimplySim: Apartamento, fábrica, casa moderna, ao ar livre, urbano. Várias aplicações foram adicionadas ao conjunto, como Simulator Maze, ou Simulação de Futebol, que é desenvolvido pela Microsoft. O sensor Kinect pode ser usado em um robô no ambiente RDS. O RDS também inclui um sensor de Kinect simulado. Os Serviços Kinect para RDS são licenciados para uso comercial e não comercial. Eles dependem do Kinect para Windows SDK . [9]

Figura 3 - Digrama da linguagem VPL.



Fonte: [10]

3 RESULTADOS E DISCUSSÕES

A partir da pesquisa e da análise realizadas foram determinados itens importantes para serem levados em consideração. Através destes foi criada a tabela comparativa a seguir.

Tabela 1 - Comparação das linguagens de programação.

LINGUAGENS DE PROGRAMAÇÃO	LEGO MINDSTORM NXT	MATLAB	MICROSOFT ROBOTICS DEVELOPER STUDIO (MSRS)
SIMULADOR	Não	Não	Sim
CONEXÃO VIA BLUETOOTH	Sim	Sim	Sim
CONEXÃO VIA USB	Sim	Sim	Não
TIPO DE LINGUAGEM	Gráfica	Gráfica	Gráfica
WINDOWS	Sim	Sim	Sim
LINUX	Não	Sim	Não
Mac OS X	Sim	Sim	Não

Fonte: Os autores.

4 CONCLUSÃO

Podemos concluir diante desse artigo comparativo que as linguagens de programações possuem poucas, mas significativas diferenças, das três linguagens citadas na pesquisa podemos perceber quais são elas. Esse artigo nos proporcionou o conhecimento mais a fundo, onde foram apresentados as suas definições e funções. Os resultados da tabela nos oferecem um fácil

entendimento e nos auxiliam na escolha de uma linguagem na hora do desenvolvimento de um projeto.

REFERÊNCIAS

- [1] CEREGATTI, Deise: Desenvolvimento de Robótica Inteligente em Ambiente Virtual para Múltiplos Robôs. Disponível em: [http://www.pucrio.br/pibic/relatorio_resumo2012/relatorios_pdf/ctc/ELE/ELEDei se%20Regin %20Ceregatti%20Momm.pdf](http://www.pucrio.br/pibic/relatorio_resumo2012/relatorios_pdf/ctc/ELE/ELEDei%20Regin%20Ceregatti%20Momm.pdf)
- [2] Disponível em: <http://masterrobot.blogspot.com.br/2010/03/lego-mindstorms.html>
- [3] ESTEVES Bruno, FURTADO Geraldo, PEREIRA Rondineli, OLIVEIRA Lindolpho: Utilização do Kit Lego Mindstorm NXT No Ensino de Controle de Processos. Disponível em: <http://www.abenge.org.br/CobengeAnteriores/2012/artigos/104237.pdf>
- [4] Disponível em: <http://www.mathworks.com/help/simulink/ug/open-the-block-library-for-lego-mindstorms-nxt-hardware.html>
- [5] Matlab, Math Works. Disponível em: <http://www.mathworks.com/products/matlab/?nocookie=true>
- [6] POSTAL, A.; PAETZOLD, G. H.; CASTRO, J. P.; JESUINO, T. Z. Controles Remotos
- [7] Disponível em: [http://i.msdn.microsoft.com/Bb483094.CompletedAction\(en-us,MSDN.10\).png](http://i.msdn.microsoft.com/Bb483094.CompletedAction(en-us,MSDN.10).png)
- [8] RDS, Microsoft Robotics Developer Studio. Disponível em: <http://www.microsoft.com/robotics/>
- [9] Visual Programming Language, Microsoft MSDN. Disponível em: <http://msdn.microsoft.com/en-us/library/bb483088.aspx>
- [10] Disponível em: <http://www.google.com.br/search?um=1&hl=pt-BR&tbm=isch&q=nxt+mindstorm&spell=&sa=X&ei=CR0JUuDNLrO64AP0voBw&ved=0CE4QvwUoAA&biw=1600&bih=756>